

# **GForge Automated Work Reviews**

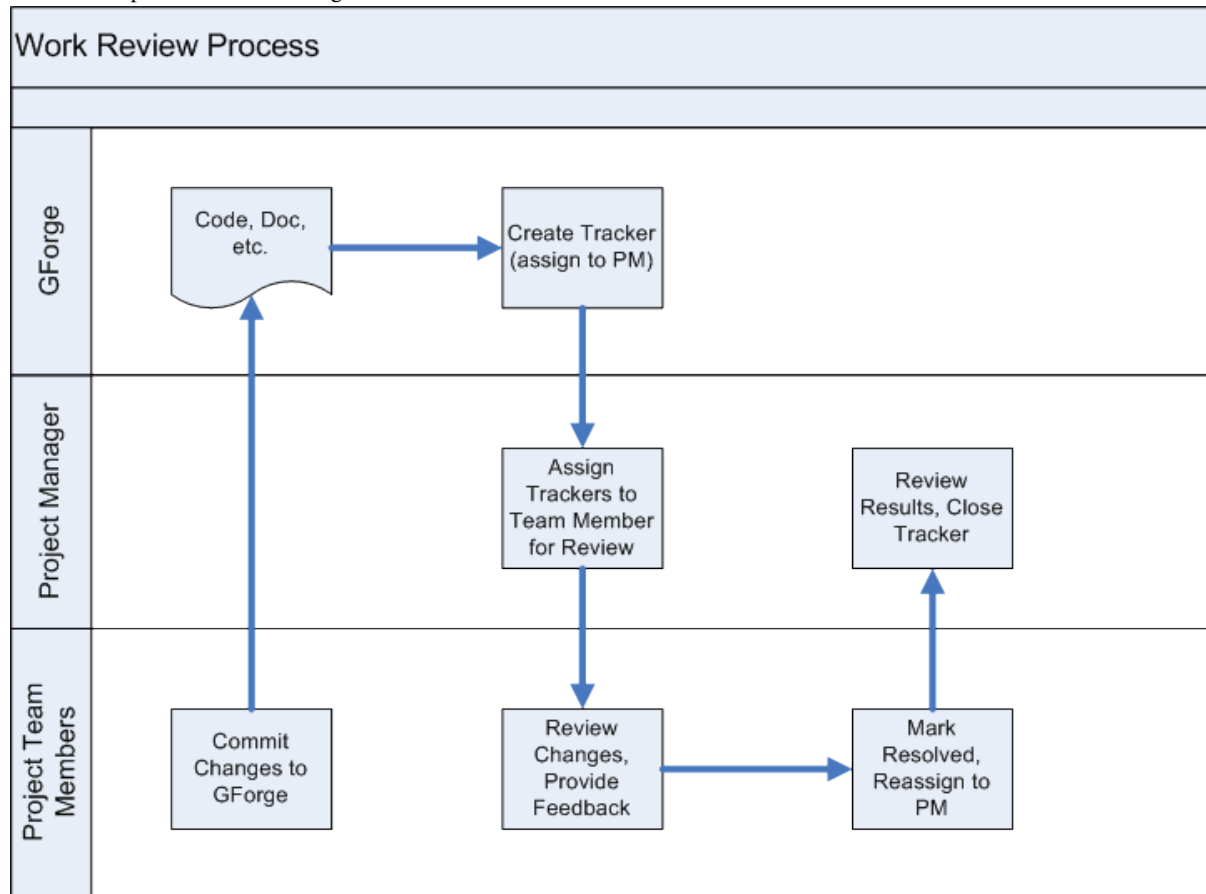
Mike Phillips  
2014/07/01 13:51

# Table of Contents

Review Process Steps .....	3
Roles & Responsibilities .....	3
Setting Up Work Reviews .....	4
What Are We Looking For? .....	4

The GForge system can automatically create new Tracker items based on code and other artifacts being committed to version control. The created Tracker items can be used to manage a peer-review process, and make sure that all work products get the right level of scrutiny.

The basic process is shown at right.



### Review Process Steps

1. A team member commits code, documents or other deliverables to the source control for a GForge project.
2. GForge stores the update, and generates a difference comparison with the previous version in HTML for each file that was updated. Diffs are only produced for text documents (like source code, config files, etc.), not binary files (like pictures, Word docs, etc.)
3. A background GForge process picks up all of the comparison files and creates a tracker for each *set* of changes submitted at one time by a given user. The comparison file is added to the tracker as an attachment.
4. The Project Manager (or anyone else on the project team, for that matter) browses the list of "Work Review" trackers, and assigns new ones to team members for review.
5. Team members review the trackers that are assigned to them, make comments, follow up with the submitter on any questions or issues they find.
6. After this follow up, the reviewer re-assigns the Tracker item to the Project Manager.
7. The Project Manager checks the outcome of each review, and closes the Tracker item.

### Roles & Responsibilities

#### Project Team Members:

- Submit changes to version control
- Include comments and bug #'s related to each commit
- Review and comment on other team members' commits as assigned by PM

#### Project Managers:

- Assign reviews promptly to catch problems quickly

- Review comments made by reviewers before closing

## Setting Up Work Reviews

To use automated work reviews, each GForge project must have a Tracker called "Work Reviews". The background process will check for this tracker and submit all project changes to it if it exists.

If your project does not have this Tracker, any project administrator (or GForge system administrator) can add it. No special fields are required within the Tracker, only the name is significant. Any other fields can be added based on project needs, and these will be ignored by the automation process.

## What Are We Looking For?

That's a great question, and obviously could be the subject of much discussion. Here are the basics:

### Consistency with established process

1. Standard documents, document titles, and organization of deliverables.
2. Make more frequent, small commits. It's best to commit one "unit of work" at a time - one bug fix, one feature or project task.
3. Don't be afraid to commit code or documents in an intermediate stage. Try not to commit code that won't compile, or that has major problems.

### Appropriate use of app frameworks

1. Use components that are provided in the framework ([PHP](#), [ASP.NET](#), [Java](#))
2. Copy, paste and update code wherever possible - don't re-invent the wheel
3. Observe naming conventions that are in place for the project. These may change over time, but consistency within the project is more important than between projects.

### General coding practices

1. Design and (where possible) test code for common security vulnerabilities: [CSRF](#), [XSS](#), [Injection](#), etc.
2. Add comments anywhere that a "trail of breadcrumbs" would be helpful to the next person
3. Add unit tests wherever possible. Even if the tests seem trivial, making the code testable almost always improves its structure.

*Note: Please feel free to add your own review criteria. Keep in mind that everything added to the wiki is subject to merciless editing and revision, so leave your ego at the door.*